



(12) 发明专利

(10) 授权公告号 CN 108777676 B

(45) 授权公告日 2021. 01. 12

(21) 申请号 201810436421.7

(22) 申请日 2013.04.25

(65) 同一申请的已公布的文献号
申请公布号 CN 108777676 A

(43) 申请公布日 2018.11.09

(30) 优先权数据
61/638,048 2012.04.25 US

(62) 分案原申请数据
201380025465.6 2013.04.25

(73) 专利权人 三星电子株式会社
地址 韩国京畿道

(72) 发明人 朴勍模 柳诚烈 黄承吾 宋在涓

(74) 专利代理机构 北京市柳沈律师事务所
11105

代理人 蔡军红

(51) Int.Cl.
H04L 29/06 (2006.01)
H04N 21/236 (2011.01)

(56) 对比文件
CN 1497876 A, 2004.05.19
CN 101563890 A, 2009.10.21
US 2010226366 A1, 2010.09.09

审查员 陈丽锋

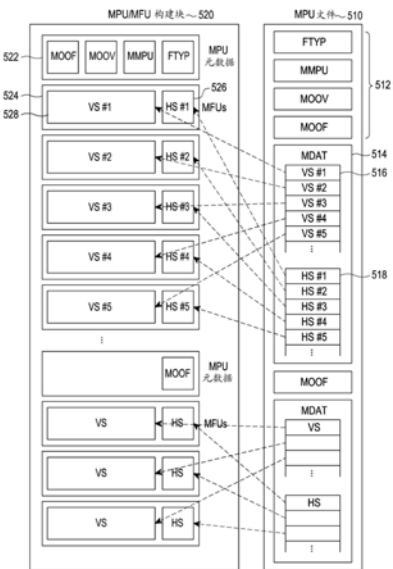
权利要求书2页 说明书12页 附图3页

(54) 发明名称

用于在多媒体传输系统中接收媒体数据的装置及方法

(57) 摘要

公开了用于多媒体传输系统的收发数据的方法和装置。本发明的方法包括步骤：从封装功能层输入包括一个或多个第一数据单元的第二数据单元；分析第二数据单元的报头信息，以便形成包括有效载荷类型和成分信息的有效载荷报头信息；根据第一数据单元的大小，形成来自第二数据单元的一个或多个有效载荷；组合与每个有效载荷相关的多条有效载荷报头信息，以便形成多媒体数据分组；以及将多媒体数据分组发送到对方实体。有效载荷包括至少一个有效载荷元素，所述至少一个有效载荷元素的类型是从包括一个第一数据单元的第一类型、包括一个第一数据单元的报头信息的第二类型、包括一个第一数据单元的报头信息和数据的一部分或者仅包括数据的一部分的第三类型、包括第二数据单元的报头信息的第四类型、和包括第二数据单元的全部的第五类型当中选择的。



1. 一种用于在多媒体传输系统中接收媒体数据的方法,所述方法包括:

接收一个或多个多媒体数据分组,每一个多媒体数据分组包含分组报头和有效载荷;
和

从所生成的一个或多个多媒体数据分组获取被分段成一个或多个子数据单元的数据单元,

其中,所生成的一个或多个多媒体数据分组中的每一个包括类型字段,所述类型字段指示在有效载荷中包含的各个有效载荷数据是由数据单元的元数据构成还是包含从所述一个或多个子数据单元得出的数据元素,

其中,如果各个有效载荷数据由数据单元的元数据构成,则所述类型字段被设置为第一值,该第一值指示各个有效载荷数据由数据单元的元数据构成,以及

其中,如果有效载荷数据包含从所述一个或多个子数据单元得出的数据元素,则所述类型字段被设置为第二值,该第二值指示各个有效载荷数据包含从所述一个或多个子数据单元得出的数据元素。

2. 如权利要求1所述的方法,其中,所述数据单元的元数据包括关于所述一个或多个子数据单元的配置的信息。

3. 如权利要求1所述的方法,其中,所述数据单元的元数据包括:具有编解码器配置信息的MOOV框、文件类型信息、和放置在所述文件类型信息之后以提供关于所述数据单元的信息的MMPU框。

4. 如权利要求1所述的方法,其中,所述类型字段包含在各个多媒体数据分组的分组报头中。

5. 如权利要求1所述的方法,其中,所述有效载荷包括:指示定时信息是否包含在所述有效载荷数据中的字段。

6. 一种用于在多媒体传输系统中接收媒体数据的装置,所述装置包括:

接收单元,被配置为接收一个或多个多媒体数据分组,每一个多媒体数据分组包含分组报头和有效载荷;和

处理单元,被配置为从所生成的一个或多个多媒体数据分组获取被分段成一个或多个子数据单元的数据单元,

其中,所生成的一个或多个多媒体数据分组中的每一个包括类型字段,所述类型字段指示在有效载荷中包含的各个有效载荷数据是由数据单元的元数据构成还是包含从所述一个或多个子数据单元得出的数据元素,

其中,如果各个有效载荷数据由数据单元的元数据构成,则所述类型字段被设置为第一值,该第一值指示各个有效载荷数据由数据单元的元数据构成,以及

其中,如果有效载荷数据包含从所述一个或多个子数据单元得出的数据元素,则所述类型字段被设置为第二值,该第二值指示各个有效载荷数据包含从所述一个或多个子数据单元得出的数据元素。

7. 如权利要求6所述的装置,其中,所述数据单元的元数据包括关于所述一个或多个子数据单元的配置的信息。

8. 如权利要求6所述的装置,其中,所述数据单元的元数据包括:具有编解码器配置信息的MOOV框、文件类型信息、和放置在所述文件类型信息之后以提供关于所述数据单元的

信息的MMPU框。

9.如权利要求6所述的装置,其中,所述类型字段包含在各个多媒体数据分组的分组报头中。

10.如权利要求6所述的装置,其中,所述有效载荷包括:指示定时信息是否包含在所述有效载荷数据中的字段。

用于在多媒体传输系统中接收媒体数据的装置及方法

[0001] 本申请是申请日为2013年04月25日、申请号为201380025465.6、发明名称为“用于多媒体传输系统的收发数据的方法和装置”的发明专利申请的分案申请。

技术领域

[0002] 本公开涉及多媒体传输系统,而且更具体地,涉及用于发送和接收多媒体数据的方法和装置。

背景技术

[0003] 术语“多媒体服务”是指诸如视频通话的呼叫服务、诸如视频点播(VOD)服务的流媒体服务、或者诸如组播和广播服务的服务。实时多媒体服务可以依赖于服务类型被划分为呼叫服务、交互式服务和流传输服务,并且也可以依赖于参与用户的数目被划分为单播、组播和广播。

[0004] 传统的广播网络通常使用运动图像专家组-2传输流(MPEG-2TS),来对多媒体内容进行传输。MPEG-2TS被用作在充斥错误的传输环境中发送比特流的典型的传输技术,利用该技术复用多个广播节目(或多个编码的视频比特流)。MPEG-2TS适合用于多媒体时代中的数字TV广播等。

[0005] MPEG-2TS可以在支持多媒体服务方面存在一些限制。换句话说,MPEG-2TS可以具有诸如单向通信、由于固定帧大小导致的传输低效率、以及在使用特定于音频/视频的传输协议和网际协议(IP)发送数据时发生的不必要的开销的限制。

[0006] 因此,MPEG最新提出MPEG媒体传输(MMT)标准作为用于支持基于MPEG技术的多媒体服务的多媒体传输技术之一。例如,MMT标准可以被应用于通过异构网络有效发送混合内容。如本文所用的术语“混合内容”可以指具有与视频/音频/应用相对应的多媒体元素的内容的集合。异构网络可以是,例如,广播网络和移动通信网络共存于其中的网络。

[0007] MMT标准旨在定义IP友好的传输技术,其是用于多媒体服务的传输网络中的基本技术。为此,需要MMT传送分组的结构,以用于在改变的多媒体服务环境中更有效地传递包括编码的媒体数据的各种格式的逻辑数据对象。

发明内容

[0008] 技术问题

[0009] 本公开的实施例的一方面提供了一种在多媒体传输系统中有效地发送和接收编码的媒体数据的方法和装置。

[0010] 本公开的实施例的另一方面提供了一种在多媒体传输系统中发送和接收各种格式的逻辑数据对象的方法和装置。

[0011] 本公开的实施例的另一方面提供了一种在支持基于IP的多媒体服务的广播系统中配置异构多媒体数据分组的方法和装置。

[0012] 技术方案

[0013] 根据本公开的实施例的一方面,提供了一种在多媒体传输系统中发送数据的方法。该方法包括:从封装功能层接收具有一个或多个第一数据单元的第二数据单元;通过分析第二数据单元的报头(header)信息,配置包括有效载荷类型和配置信息的有效载荷报头信息;基于第一数据单元的大小,配置来自第二数据单元的一个或多个有效载荷;通过组合与每个有效载荷相对应有效载荷报头信息来配置多媒体数据分组;以及将多媒体数据分组发送到对方实体。

[0014] 根据本公开的实施例的另一方面,提供了一种在多媒体传输系统中接收数据的方法。该方法包括:从对方实体接收具有有效载荷、以及包括有效载荷类型和配置信息的有效载荷报头信息的多媒体数据分组;以及从多媒体数据分组检测由封装功能层生成的一个或多个第一数据单元、以及具有所述一个或多个第一数据单元的第二数据单元的至少一部分。

[0015] 根据本公开的实施例的再一方面,提供了一种在多媒体传输系统中发送数据的装置。该装置包括:数据生成单元,被配置为输出具有一个或多个第一数据单元的第二数据单元;数据配置单元,被配置为通过分析第二数据单元的报头信息来配置包括有效载荷类型和配置信息的有效载荷报头信息,基于第一数据单元的大小来配置来自第二数据单元的一个或多个有效载荷,通过组合与每个有效载荷相对应有效载荷报头信息来配置多媒体数据分组;以及数据传输单元,被配置为将多媒体数据分组发送到对方实体。

[0016] 根据本公开的实施例的又一方面,提供了一种在多媒体传输系统中接收数据的装置。该装置包括:接收单元,被配置为从对方实体接收具有有效载荷、以及包括有效载荷类型和配置信息的有效载荷报头信息的多媒体数据分组;以及解释单元,被配置为从多媒体数据分组检测由封装功能层生成的一个或多个第一数据单元、以及具有所述一个或多个第一数据单元的第二数据单元的至少一部分。

[0017] 有效载荷可以包括与包括一个第一数据单元的第一类型、包括一个第一数据单元的报头信息的第二类型、包括一个第一数据单元的报头信息和数据的一部分或者仅包括数据的一部分的第三类型、包括第二数据单元的报头信息的第四类型、和包括第二数据单元的全部的第五类型中的任一个相对应的至少一个有效载荷元素。

附图说明

[0018] 图1示出了根据本公开的实施例的MMT系统的分层结构。

[0019] 图2示出了根据本公开的实施例的MMT有效载荷的结构。

[0020] 图3示出了根据本公开的实施例的MMT包的逻辑结构。

[0021] 图4示出了根据本公开的实施例的在MMT系统中配置和发送包含MMT有效载荷的多媒体数据分组的流程图。

[0022] 图5示出根据本公开的实施例的生成多媒体数据分组的操作。

具体实施方式

[0023] 现在,将参照附图详细描述本公开的优选实施例。在本公开的以下描述中,当并入本文的公知的功能和配置的详细描述可能模糊本公开的主题时将被省略。本文所用的术语是通过考虑其在本公开中的功能而定义的,并可以根据用户、操作者的意图或通常的实践

而改变。因此,该定义应当基于遍及整个说明书的内容来做出。

[0024] 将在下文描述在MMT标准中定义的分层结构。本公开的实施例提供了多媒体数据分组的结构。为此,将详细描述MMT标准中定义的分层结构中的用于生成多媒体数据分组的层。如本文所用的术语“多媒体数据分组”可以指用于MMT服务的传递帧。

[0025] MMT协议(MMTP)定义了用于通过IP网络传递被配置在MMT有效载荷格式(PF)中的传递帧的应用层协议。MMT有效载荷被设计为通过被配置在MMT有效载荷格式中来有效地传递。为了通过异构IP网络有效地传递MPEG媒体数据,MMT定义了封装格式、传递协议和信令消息格式。

[0026] 图1示出了根据本公开的实施例的MMT系统的分层结构。

[0027] 参照图1,示出了媒体编码层110、封装功能层(层E)120、传递功能层(层D)130、传送协议层140、网际协议(IP)层150、和控制功能层(层C)100,它们用于配置多媒体数据分组和发送经配置的多媒体数据分组。

[0028] 按照本公开的实施例,媒体编码层110和封装功能层120可以作为多媒体数据生成单元进行操作,该多媒体数据生成单元用于生成基于多媒体内容和/或多媒体服务的多媒体数据。传递功能层130可以作为多媒体数据配置单元进行操作,该多媒体数据配置单元用于基于从多媒体数据生成单元接收到的多媒体数据来配置多媒体数据分组。与多媒体数据配置单元相对应的传递功能层130可以通过识别从多媒体数据生成单元提供的至少一个多媒体数据来配置报头信息,并通过组合报头信息和至少一个多媒体数据来配置多媒体数据分组。

[0029] 在媒体编码层110中压缩的多媒体数据可以通过封装功能层120以类似于文件格式的形式被分包(packetize)。封装功能层120可以通过接收从媒体编码层110提供的编码的媒体数据或者存储的媒体数据来生成数据段,它是MMT服务的小单元,并且可以使用数据段生成MMT服务的访问单元(access unit)。封装功能层120可以通过组合和/或划分访问单元来生成用于生成/存储和传输混合内容的分组格式。

[0030] 传递功能层130可以将封装功能层120输出的(多个)数据单元转换为MMT有效载荷格式,然后将MMT传送数据分组报头添加到MMT有效载荷格式以配置MMT传送数据分组,或者可以使用现有的传送协议,实时协议(RTP),来配置RTP分组。

[0031] 在传递功能层130中配置的数据分组可以在经历诸如用户数据报协议(UDP)或传送控制协议(TCP)的传送协议层140之后,最后在IP层150中被IP分包(IP-packetized)。传送协议层140和IP层150可以作为数据传输单元进行操作。控制功能层100,它是可选的,可以生成数据传输所需的控制信息或信令信息,将所生成的信息添加到数据,并发送数据,或者可以通过单独的信令手段发送数据。

[0032] 在传递功能层130中生成的MMT有效载荷格式可以定义将通过MMT协议或RTP传递的(多个)媒体单元的逻辑结构。MMT有效载荷可以由如下有效载荷格式来指定:该有效载荷格式通过MMT层协议或其他现有应用的传输协议来传递封装的数据单元或其它信息。MMT有效载荷可以提供关于流传输的信息和关于文件传输的信息。在流传输中,数据单元可以是MMT媒体片段单元(MFU)或MMT处理单元(MPU)。在文件传输中,数据单元可以是MMT资源(asset)和MMT包(package)。

[0033] 图2示出了根据本公开的实施例的MMT有效载荷的结构。

[0034] 参照图2,MMT有效载荷200可以被配置为包括至少一个MMT MFU210、至少一个MMT MPU 220、至少一个MMT资源230、以及至少一个MMT包240中的至少一个。

[0035] MFU 210可以是包括编码的媒体数据的通用容器格式,它是独立于任何媒体编解码器,并且可以由媒体解码器独立处理。MFU 210,它代表MPU 220的分段数据的一部分,可以是能够被独立解码的最小单元。作为一个例子,如果通过使用一个帧作为访问单元来执行编码,则MFU 210可以是一个视频帧。在另一情况下,MFU 210可以是包含在一个帧中的一个切片(slice)。

[0036] MPU 220,它是包括一个或多个MFU以及附加的传递和处理相关的信息的容器格式,可以包括从多个不同的访问单元生成的各种数目的MFU。MPU 220,它指能够完全和独立地被MMT兼容实体处理的编码的媒体数据单元,依赖于应用环境可以具有特定大小(例如,在视频的情况下,一个画面组(one Group of Picture,GOP))。作为一个例子,MPU 220可以具有(或包括)构成一个GOP(例如,1秒视频)的多个画面帧,并且MFU 210可以包括每个画面帧。

[0037] MMT资源230,它是具有一个或多个MPU的数据实体,可以是相同成分信息和传送特性被应用到其中的最大数据单元。MMT资源230可以仅包括一种类型的数据,包括分包的或复用的数据。例如,音频的基本流(ES)的至少一部分、视频的ES的至少一部分、MPEG-UI(用户界面)窗口小部件包的至少一部分、MPEG-2传送流(TS)的至少一部分、MPEG-4(MP4)文件的至少一部分、并且MMT包的全部或至少一部分都可以是MMT资源230。

[0038] ES,它由特定媒体编解码器定义,在逻辑上可以是一个或多个MMT资源。支持分层编解码器和多视图编解码器的MMT资源230可以与其他MMT资源重叠。

[0039] MMT成分信息(MMT-CI)指定MMT资源的空间和时间关系的信息,而且MMT传送特性(MMT-TC)定义了传递MMT资源所要求的服务质量(QoS)。对于特定传递环境,MMT-TC可以被表达为资源传递特性(ADC)。

[0040] MMT包240可以被定义为由MMT兼容实体处理的编码的媒体数据及其相关信息的集合。

[0041] 图3示出了根据本公开的实施例的MMT包的逻辑结构。如图所示,MMT包300可以具有(和/或包括)一个或多个MMT资源320、MMT成分信息310、以及代表MMT传送特性的一个或多个ADC 330。MMT包300(其可与图2中所示的MMT包240类似)可以包括诸如标识符的描述信息,和MMT资源320的位置,而且MMT包300中的MMT资源320可以被复用和/或级联。

[0042] MMT包300可以以MPU为单位进行处理,并且MMT资源320可以是具有相同MMT资源ID的一个或多个MPU的集合。与每个MMT资源320有关的传送特性可以由ADC 330来表示。ADC 330可以被对MMT包进行分包的实体用来配置MMT有效载荷的参数以及下面描述的MMT包的报头信息。

[0043] 在可以包括在MMT包的逻辑实体当中,MPU可以包括基于给定定义的报头字段,而且这些字段中的一些对处理MPU而言可能是必要的。在MPU的有效载荷中的MFU的边界外的一部分被切除的情况下,如果分组丢失,则错误可能传播更长时间。在要求低时延的媒体数据中,MFU需要在其生成时立即发送。在这种情况下,为了防止传输延迟,可以将传输顺序确定为:在发送所生成的MFU数据之后发送MPU的报头字段信息。

[0044] 鉴于上述特性,MMT有效载荷需要被配置为保持MPU数据分组和MFU分组之间的兼

容性。

[0045] 图4示出了根据本公开的实施例的在MMT系统中配置和发送包含MMT有效载荷的多媒体数据分组的流程图。图4中所示的操作可以通过图1中的层当中的传递功能层130来执行。

[0046] 参照图4,在操作410中,传递功能层130可以接收从封装功能层120提供的多媒体数据。封装功能层120可以提供配置多媒体数据分组所必须的多媒体数据、以及与每个多媒体数据有关的信息。

[0047] 在操作420中,传递功能层130可以基于多媒体数据的报头信息,配置与多媒体数据分组的MMT有效载荷有关的有效载荷报头信息。作为一个例子,有效载荷报头信息可以提供MMT有效载荷的长度、有效载荷数据的类型、以及与分段和聚集有关的信息,而且有效载荷报头格式可以由系统操作员或协议标准来定义。

[0048] 在操作430中,传递功能层130可以生成用于传递多媒体数据的多媒体数据分组的有效载荷。换句话说,传递功能层130可以通过将在操作410中接收到的多媒体数据中的至少一部分与在操作420中配置的有效载荷报头信息进行组合,来配置MMT有效载荷。

[0049] 在操作440,MMT分组,它是包括MMT有效载荷的多媒体数据分组,可以通过给定的传送协议发送到对方实体(例如,接收器)。

[0050] 现在将描述在本公开的实施例中提出的由负责MMT服务的多媒体数据的传输的传递功能层来配置多媒体数据分组的详细过程,而且还将描述有效载荷格式。具体地,可以提供基于在封装功能层120中生成的数据的类型的传送数据结构。对于多媒体数据分组的配置,可以根据给定格式生成将被记录在报头区域中的有效载荷报头信息和将被记录在有效载荷区域中的多媒体数据。

[0051] 图5示出根据本公开的实施例的生成多媒体数据分组的操作。

[0052] 参照图5,MPU文件510指封装功能层期望发送的数据,而且MPU/MFU构建块520具有用于配置传输数据的数据结构。MPU文件510可以包括:报头信息512,诸如指示文件的类型的文件类型(FTYP)框、指示MPU的结构 MMPU框、指示编解码器设置信息的MOOV框、和电影片段(MOOF)框;以及具有视频样本(VS) 516和MMT提示样本(HS) 518的MDAT框514。MMPU框指示存储在MDAT框中的数据的类型为MPU,而且MOOV框包含关于存储在MDAT框中的媒体帧的信息。MPU/MFU构建块520可以包括诸如FTYP、MMPU、MOOV和MOOF的MPU元数据522,以及具有相关HS 526和VS 528的MFU 524。

[0053] MMT有效载荷可以基于MPU/MFU构建块520来配置,并且在MMT分组中携带或发送。

[0054] 用于配置MMT有效载荷的详细过程如下。

[0055] 数据配置单元可以分析从数据生成单元提供并且期望被发送的MPU文件形式的数据510,以从MPU数据510的报头信息512确定类型信息、MPU的配置信息、编解码器的设置信息、数据的开始点和大小(或长度)信息。数据配置单元可以分析构成MPU数据510的MFU(相应于图5中的VS) 516和MFU HS 518的结构,以确定包含每个MFU的大小、相互重要性和互连(长度等)信息的MFU报头以及作为实际媒体数据的MFU数据,并且还可以通过指示每个MFU的起始偏移和长度的MFU HS 518来分析每个MFU的大小。

[0056] 基于分析的MPU格式,可以对用于配置传送信息分组的MPU/MFU构建块520进行配置。MFU HS 518可以是MPU/MFU构建块520的MFU报头。

[0057] 数据配置单元可以基于从MFU HS 518分析的每个MFU的大小,将每个MFU配置为传送单元。在这种情况下,依赖于MFU的大小,可以配置一个或多个有效载荷。每个有效载荷的最初、中间或最后部分可以由start_end_indicator (它是可选的比特) 来指示。MPU元数据522还可以被视为MPU/MFU构建块520的一个配置单元。

[0058] 每个有效载荷元素的数据类型可以呈现如下。

[0059] i. 具有FTYP、MMPU、MOOV、MOOF和MDAT的一个有效载荷522的类型被称为MPU_Partial(),并且可以被配置为仅包含MPU报头。

[0060] ii. 如果一个MFU被配置为一个有效载荷,则有效载荷的类型被称为MFU_set()。

[0061] iii. 在一个MFU被配置为多个有效载荷的情况下,如果MFU仅包括MFU报头,则每个有效载荷的类型是MFU_Partial(),并且如果MFU包括MFU报头和MFU数据的一部分,或者包括MFU数据的一部分,则每个有效载荷的类型是MFU_chunk()。

[0062] iv. MPU_Partial()可以指仅包括MPU报头的有效载荷的类型。

[0063] v. MPU_Set()可以指完全包括MPU的有效载荷的类型。

[0064] 下面示出了根据本公开的实施例的MMT有效载荷的语法。下面的MMT有效载荷格式可以用于UDP传送,点对点组块(chunk)的交换,以及文件的传送。

```
MMT_Payload()
{
```

```
    bit(N) type;
    bit(N) asset_id;
    switch (type) {
    case 0:
        //配置;
        if (asset_id==0) {
            //包配置数据
        } else {
            //资源配置数据
        }
        break;
    case 1:
        MFU_Partial();
        break;
[0066] case 2:
        MFU_chunk();
        break;
    case 3:
        MFU_set ();
        break;
    case 4:
        MPU_Partial();
        break;
    case 5:
        MPU_Set();
        break;
    ...
    }
}
```

[0067] 如上所述,MMT有效载荷可以包括指示有效载荷的数据类型的类型字段,以及指示有效载荷(MPU或MFU)所属的资源的标识符的asset_id。asset_id可以存在以复用MMT比特

流。如果MMT有效载荷通过RTP传递而且使用了单个资源,则asset_id可以省略。

[0068] 依赖于类型字段,元素的下列各种情况可以被包含在MMT有效载荷中。

[0069] 如果MMT有效载荷包含将被发送的数据的配置数据,则可以使用情况0。如果asset_id=0,则MMT有效载荷可以包括包配置数据。否则,资源配置数据可以被包括。

[0070] 当MMT有效载荷携带MFU数据的配置数据(报头信息)时,可以使用与情况1相对应的MFU_Partial()元素。当MMT有效载荷包含MFU数据的一部分连同MFU数据的配置数据(报头信息)或者携带MFU数据的全部或一部分时,可以使用与情况2相对应的MFU_chunk()元素。当MMT有效载荷包含MFU数据的全部(报头和数据)时,可以使用与情况3相对应的MFU_set()元素。当MMT有效载荷包含MPU的数据时,可以使用与情况4相对应的MPU_Partial()元素。当MMT有效载荷包含MPU数据的全部(报头和数据)时,可以使用与情况5相对应的MPU_Set()元素。

[0071] 此外,MMT有效载荷还可以包括其他单元,例如文件单元、控制层的信令信息和保护消息。

[0072] 当MFU的大小超过最大传输单元(MTU)的大小时,MFU_Partial()可以被定义为:被添加到有效载荷字节的起始处并且被定义在MFU中的所有报头的集合。

[0073] MFU_chunk()可以被定义用于在MFU_Partial()分组中发送的报头的、MFU的有效载荷的中间或最后部分。

[0074] MFU_set()可以被定义为包括预定数目的完整MFU的分组。

[0075] MPU_Partial()可以被定义为被添加到附加MFU_Partial()元素的所有报头的集合,所述附加MFU_set()元素被添加到可选的MFU_Partial()元素并且在MPU中定义。可以依赖于分组的剩余字节和MFU长度来确定附加元素的存在/不存在。

[0076] 现在将描述MPU和MFU元素的例子以及MMT有效载荷格式和MMT传送分组之间的映射。本文将省略本领域已知的或与本公开的实施例无关的参数字段的描述,但是应注意的是,这样的参数字段不限制本公开的范围。

[0077] 以下代表MFU元素的一个例子。

mmt_mfu_element()

{

...

optional bit(2) start_end_indicator;

[0078] *optional bit(1) rap_fragment_indicator;*

optional bit(N) layer_dependency_info;

optional bit(N) scrambling_key_index;

payload();

}

[0079] 在上面的例子中,start_end_indicator(起始结尾指示符)可以是可选的比特,其指示有效载荷是MFU的起始、中间或结尾。此外,start_end_indicator可以指示MFU包括随

机存取点 (RAP) 数据, 并且可以包括最佳比特, 其指示解码数据的起始和/或结尾位置。rap_fragment_indicator (rap 片段指示符) 可以是可选的比特, 如果有效载荷包含可随机访问的数据的起始点, 则 rap_fragment_indicator 指示可以直接解码的数据的起始点的地址。

[0080] 如果有效载荷被加密, 则 scrambling_key_index (加扰密钥索引) 可以包含关于能够解密加密的一系列值的信息。

[0081] layer_dependency_info (层依赖性信息) 可以是指示画面层之间的依赖性的信息。此外, 它还可以包括层标识符 (LayerID)、依赖性计数、类似于 MPEG-2TS 分组的信号滚动键 (signal rolling key) 等。

[0082] 以下代表 MPU 元素的一个例子。

```
mmt_mpu_element(multiple, timing)
{
    optional bit(2) start_end_indicator;
    bit (1) multiple_mfu_flag;
    bit (1) no_mfu
[0083] bit (1) timing_flag;
    bit(1) codec_update_flag;
    bit(1) event_list_flag;
    bit(1) clock_map;
    bit(1) headers_first;
```

```

    bit(1) multiple_aus_flag;
    if (codec_update_flag) {
        bit(16) codec_config_size;
        bytes(codec_config_size) codec_config_data;
        bit(8) downcounter ;//处理延迟 (移除)
    }
    if (event_list_flag) {
        bit(16) num_events;
        for (i=0; i<num_events; i++) {
            ... //事件数据
        }
    }
    if (multiple_mfu_flag) {
        bit(N) num_MFUs;
    } else {
[0084]     num_MFUs = 1;
        headers_first = 0;
        multiple_aus_flag = 0;
    }
    if (timing_flag) {
        optional bit(timestamps_bits) base_timestamp;
        if (clock_map) {
            bit(64) clock_ref;
        }
        optional bit(ts_inc_bits) ts_increment;
        optional bit(2) constant_flag ;
        optional bit(N) interval ;
        if (ts_inc_factor) ts_increment *= ts_inc_factor;
    }
    for (i=0; i<num_MFUs; i++) {
        optional bit(N) decode_order_number;
    }

```

```

        if (timing && (!i || multiple_aus_flag) ) {
            optional bit(N) dts_diff;
            optional bit(N) composition_timestamp_offset;
            mfu_dts = base_timestamp + dts_diff * ts_increment;
            mfu_cts = base_timestamp + composition_timestamp_offset *
ts_increment;
        }
        if (multiple_mfu_flag)
[0085]         optional bit(N) mfu_size;
        if (!headers_first) mmt_mfu_element();
    }
    if (headers_first) {
        for (i=0; i<num_MFUs; i++) {
            mmt_mfu_element();
        }
    }
}

```

[0086] 在上面的例子中, start_end_indicator可以是可选的比特, 其指示有效载荷是MFU的起始、中间或结尾。如果MPU具有多个MFU, 则multiple_mfu_flag (多mfu标记) 可以被设置为“1”。如果像在文件传输的情况下在MPU数据中没有包括MFU, 则no_mfu (无mfu) 可以被设置为“1”。

[0087] timing_flag (定时标记) 可以被用于, 在先前的分组丢失的情况下, 通过指示当前MFU是否属于与前一MFU相同的访问单元来校正时间戳。如果有效载荷具有编解码器配置信息, 则codec_update_flag (编解码器更新标记) 可以被设置为“1”。如果有效载荷开始于媒体数据的报头信息, 则header_first (开始于报头) 可以被设置为“1”。如果有效载荷具有将被呈现给用户的事件列表, 则event_list_flag可以被设置。事件列表可以列出属于内容的事件。

[0088] 如果有效载荷包含多个帧, multiple_aus_flag (多aus标记) 可以被设置为“1”。如果使用诸如高级音频编码 (AAC) 的编解码器, 则用于频带内编解码器配置的单元可以被包括在MPU元素中。

[0089] codec_update_flag可以包括指示编解码器配置信息的大小的codec_config_size (编解码器配置大小) 和指示编解码器配置信息的codec_config_data (编解码器配置数据)。在可替换的实施例中, 在多媒体流传输服务的开始时或中间, 如果编解码器基于编解码器配置信息而改变, 或者如果编解码器配置信息被修改, 则可以通过codec_update_flag来设置所述信息的存在/不存在。如果编解码器配置信息被划分成多个有效载荷, 则downcounter (递减计数器) 可以被用于指示在当前有效载荷之后存在另外一些包含编解码

器配置信息的有效载荷。作为一个例子,如果编解码器配置信息被划分为多个段而且downcounter被设置为“3”,则接收器可以确定后面还需要具有2、1和0的downcounter的有效载荷以完成编解码器配置信息。当接收到downcounter 0的有效载荷时,接收器可以确定与编解码器配置信息相关的所有有效载荷的传输已完成。

[0090] 如果event_list_flag(事件列表标记)被设置,则可以存在将要通过传输单元传递的事件列表,而且事件列表可以包括指示事件的数目的num_events(事件数目)和至少一个事件数据。在可替换的实施例,如果使用已经被发送的事件数据,则可以通过仅包括事件标识符“event ID”而不包括事件数据来表达事件。在另一可替换的实施例,可以在发送有效载荷之前使用预定的传送手段预先发送事件数据。列出的事件可以在与帧的显示同步之后执行。在可能的实施例,在播放视频数据的过程中可以可选地通过事件列表改变音频。

[0091] 如果multiple_mfu_flag被设置,则指示有效载荷具有多少个MFU的num_MFUs可以存在于MPU元素中。如果multiple_mfu_flag没有被设置,则num_MFUs被设置为“1”,而且header_first和multiple_aus_flag被设置为“0”。在可能的实施例,如果期望被发送的多媒体数据单元具有多个帧,或者在单个帧中具有多个切片,则可以设置用于设置多个实体的数目的信息。

[0092] 如果timing_flag被设置,则用于MPU中的单元的定时信息可以被包括在MPU元素中。

[0093] 此外,与num_MFUs一样多的MFU可以在MPU元素中列出。因为MFU的顺序可以不同于从解码器输入的顺序,所以可以可选地包括指示MFU被输入到解码器的顺序的decode_order_number(解码器顺序号)。

[0094] 如果multiple_mfu_flag被设置为“1”,则还可以包括指示每个MFU的大小的mfu_size(mfu大小)。在本说明书中,假定输入到数据配置单元的MFU具有相同的大小。

[0095] 如果headers_first没有被设置为“1”,则可以包括一个mmt_mfu_element()。

[0096] 如果headers_first被设置为“1”,则与num_MFUs一样多的mmt_mfu_element()可以被包括在MPU元素中。

[0097] 根据本公开的实施例配置的接收器可以包括被配置为从对方实体接收多媒体数据分组的接收单元,和被配置为依赖于MMT有效载荷的上述格式来提取和解释多媒体数据分组或至少其一部分中包括的至少一个MFU或MPU的解释单元。

[0098] MMT有效载荷的上述格式可以使能MFU或MPU的有效传送和聚集/分段。

[0099] 虽然已经参照本公开的某些示例性实施例示出和描述了本公开,但是本领域技术人员应当理解,可以在形式和细节上对其做出各种改变而不脱离由所附权利要求及其等同物定义的本公开的精神和范围。

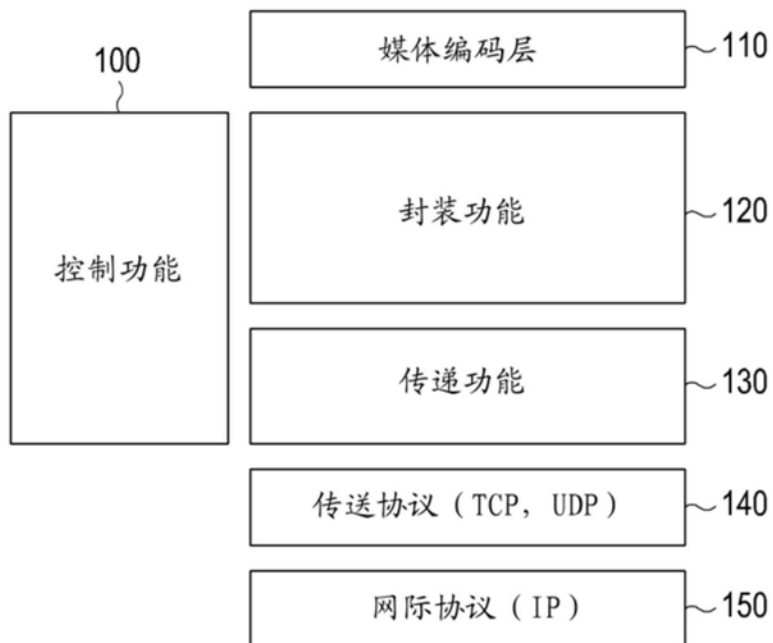


图1

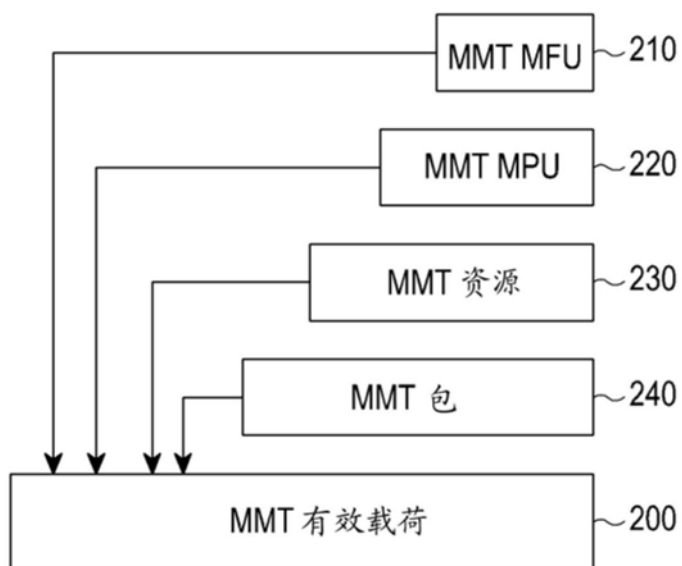


图2

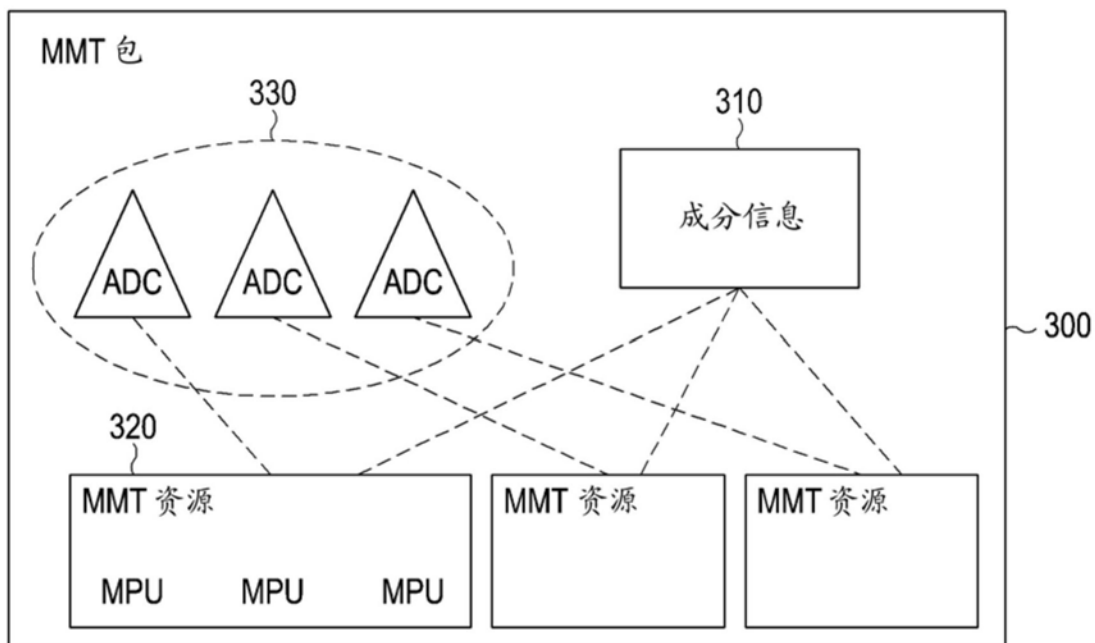


图3

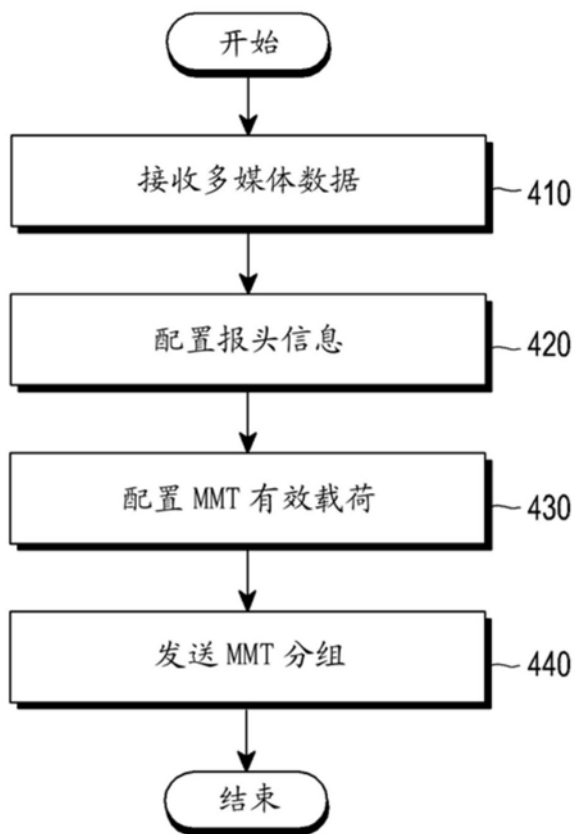


图4

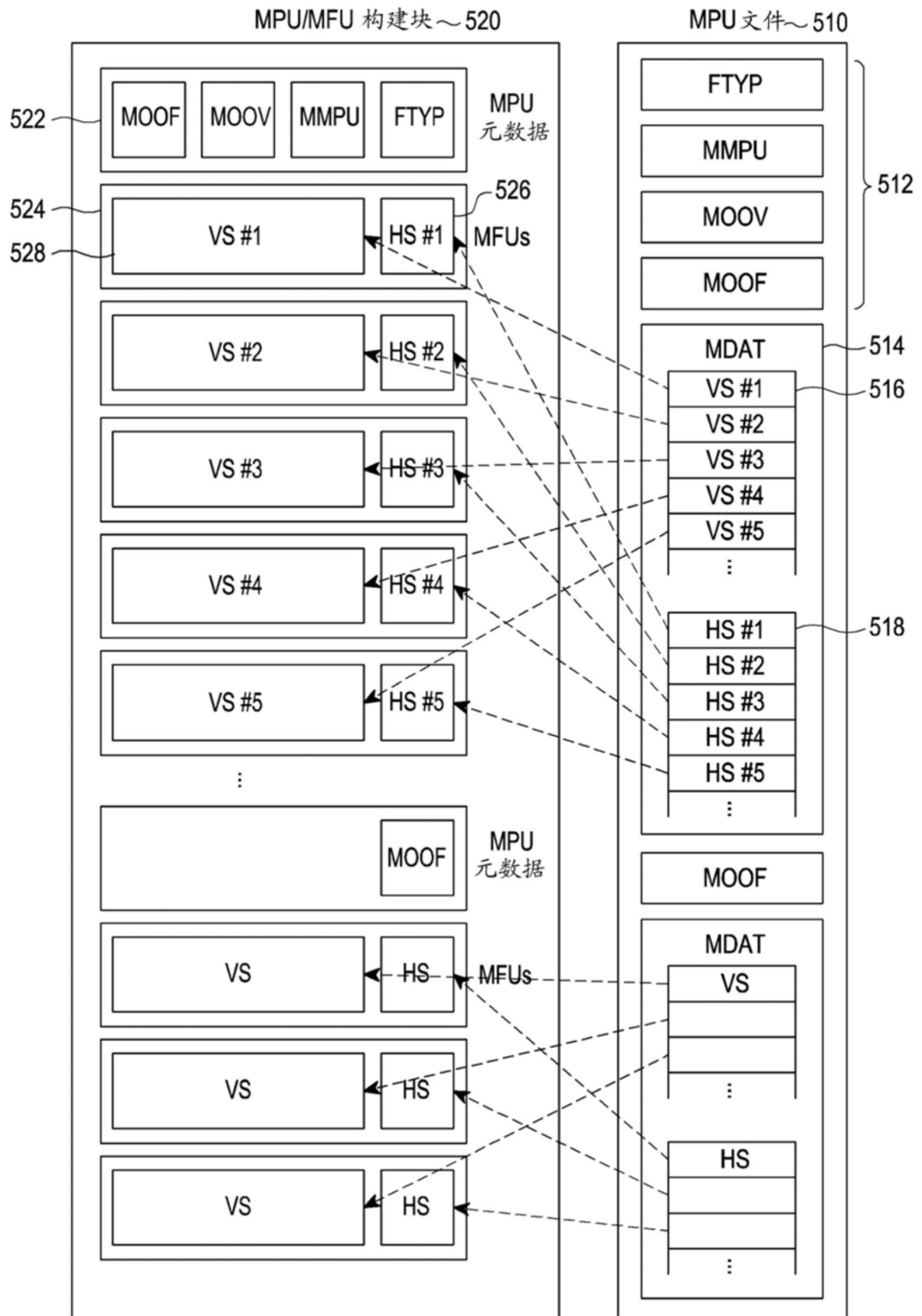


图5